Adaptive methods for multi-material ALE hydrodynamics[‡]

W. J. Rider[*, †], E. Love, M. K. Wong, O. E. Strack, S. V. Petney and D. A. Labreche

Computational Shock and Multi-physics Department, Sandia National Laboratories, MS-0378, P.O. Box 5800, Albuquerque, NM 87185-0378, U.S.A.

SUMMARY

Arbitrary Lagrangian–Eulerian (ALE) methods are commonly used for challenging problems in hydrodynamics. Among the most challenging matters are the approximations in the presence of multiple materials. The ALE code ALEGRA has used a constant volume method for computing the impact of multiple materials on both the Lagrangian step and the remap step of the method. Here, we describe modifications to these methods that provide greater modeling fidelity and better numerical and computational performance. In the Lagrangian step, the effects of differences in material response were not included in the constant volume method, but have been included in the new method. The new methodology can produce unstable results unless the changes in the variable states are carefully controlled. Both the stability analysis and the control of the instability are described. In the standard (Van Leer) method for the remap, the numerical approximation did not account for the presence of a material interface directly. The new methodology uses different, more stable and more dissipative numerical approximations in and near material interfaces. In addition, the standard numerical method, which is second-order accurate, has been replaced by a more accurate method, which is third-order accurate in one dimension. Published in 2010 by John Wiley & Sons, Ltd.

1. INTRODUCTION

The use of arbitrary Lagrangian–Eulerian (ALE) computer codes has been an enabling technology for many important defense-related applications. These computer codes are developed by combining modern algorithms for Lagrangian hydrodynamics, meshing technology and remap methods developed for high-resolution Eulerian methods. ALE methods were introduced in 1974 by Hirt, Amsden, and Cook [1], but needed the development and wide-spread use of high-resolution methods such as FCT [2], or slope-limiters [3] to become accurate enough for practical applications. This was due to the overly dissipative remap associated with the use of upwind or donor-cell differencing. The upwind type of differencing was necessary to produce physically bounded quantities in the remap essential for challenging problems.

ALEGRA began life as an arbitrary connectivity multi-material arbitrary Lagrangian Eulerian code known as MMALE. The chosen development kernel with which to start was the PRONTO code [4, 5], a Lagrangian arbitrary (but fixed) connectivity (finite element) 3-D code in existence at Sandia at the time of the planning culmination. The Lagrangian mode would accommodate changes in

length scale with a conforming mesh, while the Eulerian mode (to be added) would accommodate the complex shearing and fluid distortion associated with turbulent flows.

The proposal also regarded the new code as a potential and logical next generation extension of the CTH code [6], a production 3-D Eulerian code developed by Sandia and in wide use by the early 1990s. Although generally highly successful, CTH proved inadequate for certain classes of problems, such as capsule implosion simulations. Needless to say, a variety of new physics was required that was not implemented in either CTH or PRONTO: radiation transport, electron-ion two-temperature fluid approximations to plasma flows, electron thermal conduction, fusion burn physics, charged particle beam deposition, and potentially MHD physics associated with fast Z-pinches.

The code project which was born during the two-year period of 1988 and 1989 finally started formally in March of 1990. An early 2-D Fortran incarnation called RHALE was available later that same year. In the late fall of 1990, the decision was made to recast RHALE into an object-oriented structure using the C++ programming language [7, 8]. The advantages of this change included: enhanced data structures and memory management, object-oriented programming constructs, excellent debugging and code development environments, and improved ability to use massively parallel computing hardware. From this beginning, the full capabilities of ALEGRA were realized.

To this point, we have taken a distinctly myopic view of research in ALE methods. While our definite focus is ALEGRA, research on ALE methods is both active and high quality. Good examples of modern ALE methods can be found by the work of Shashkov [9–12] as well as in the new classical ALE methodology from Lawrence Livermore Laboratory by Tipton [13, 14]. Other extremely high-quality work can be found, but too often is not published in the readily available peer-reviewed literature. Tipton's work is also the starting point for the multi-material closure discussed below. Barlow has also contributed greatly to the closure problem utilizing an approach related to Riemann solvers [15].

The remainder of this paper is organized into four sections. In the next section, we describe the basic Lagrangian integration step including the treatment of multiple materials. The classical methodology is first described followed by our modified treatment including stability analysis and implementation. This is followed by a section describing the remap methodology both in its classical form and the modified methodology, which adapts to the presence of material interfaces. Examples are provided to demonstrate the method in use, including the verification of numerical accuracy and practical application. This paper closes with a discussion on the impact of the methods.

2. MULTI-MATERIAL LAGRANGIAN STEP

ALEGRA's Lagrangian step is based on the classical centered space and time method where the velocities are located at the grid's nodes and staggered in time and the thermodynamic variables are at element centers and staggered with respect to velocity in time. In the following $(\cdot)_j^n$ is the field $(\cdot)$ at time index n and space index j. The density ρ is implicitly computed from the node locations $\mathbf{x}$ with the nodes advancing explicitly in time using the velocity $\mathbf{v}$ and time step $\Delta t > 0$,

$$\mathbf{x}_{j+1/2}^{n+1} = \mathbf{x}_{j+1/2}^n + \Delta t \mathbf{v}_{j+1/2}^{n+1/2}. \tag{1}$$

The momentum equation is integrated as

$$\mathbf{v}_{j+1/2}^{n+1/2} = \mathbf{v}_{j+1/2}^{n-1/2} - \frac{\Delta t}{m_j} \nabla_{j+1/2} (\boldsymbol{\sigma} + \mathbf{Q}), \tag{2}$$

where $m_j > 0$ is the mass of element j. The above includes the physical stress tensor $\boldsymbol{\sigma}$ and the artificial viscosity $\mathbf{Q}$. For the sake of brevity, we will principally consider the hydrodynamic response of materials ($\boldsymbol{\sigma} = -p\mathbf{I}$, where p is the pressure and $\mathbf{I}$ is the identity tensor) and forgo the detailed description of solid mechanical response. That part of the algorithm follows the classical

description given by Wilkins [16] for a hypo-elastic material with plastic response. The energy equation follows in a similar fashion, with the internal energy density per unit mass e updated as

$$e_j^{n+1} = e_j^n - \frac{\Delta t}{m_j}(\boldsymbol{\sigma}_j + \mathbf{Q}_j) \cdot \frac{1}{2}\nabla_j(\mathbf{v}^{n-1/2} + \mathbf{v}^{n+1/2}). \tag{3}$$

The classical multi-material treatment in ALEGRA essentially does as little as possible. The basic approach is to keep the volume fractions of all materials constant through the Lagrangian step. As a result, a highly compressible material will expand or compress to the same degree as a stiff material. This is clearly unphysical, but it is simple, and it is robust as long as the materials remain in physically realizable states.

The isentropic multi-material method has been developed to allow the Lagrangian step of the ALE algorithm in ALEGRA to respond more physically. In this method, the relative magnitude of the algorithmic bulk modulus $B_k > 0$ of each material (indexed by k) determines its response to the bulk flow (expansion and compression), and pressure differences between materials. For example, should an element containing solid iron and air be compressed, the air should be compressed more than the iron. The method also allows materials to reach the same hydrodynamic pressure consistent with the fundamental assumptions for the governing equations. The multi-material equations are the same as used by Tipton (discussed in [17]) and similar to those given by Miller and Puckett [18]. In the Lagrangian frame of reference, the evolution of material volume fraction f_k is given by

$$\frac{\mathrm{d}f_k}{\mathrm{d}t} = f_k\left(\frac{\bar{B}}{B_k} - 1\right)(\nabla \cdot \mathbf{v}) - \frac{f_k}{\bar{p}}\frac{\mathrm{d}p_k}{\mathrm{d}t}. \tag{4}$$

The derivation of this equation is given by Colella *et al.* [19] and perhaps written down in print first by Tipton [14]. The energy update equation is

$$f_k\rho_k\frac{\mathrm{d}e_k}{\mathrm{d}t} = f_k\bar{p}\frac{\bar{B}}{B_k}(\nabla \cdot \mathbf{v}) - f_k\frac{\mathrm{d}p_k}{\mathrm{d}t}, \tag{5}$$

where $\rho_k > 0$ is the material density, p_k is the material pressure, e_k is the material internal energy density per unit mass, $\bar{B} > 0$ is the algorithmic mean bulk modulus and $\bar{p}$ is the algorithmic mean pressure. After applying conservation of energy and back substituting, the discrete energy update becomes

$$m_k\Delta e_k = \bar{p}\Delta f_k, \tag{6}$$

where $m_k = \rho_k f_k$ (a constant over this step), and $\Delta(\cdot)$ is the change in the variable $(\cdot)^{n+1} - (\cdot)^n$. Included are the following closure assumptions:

$$\bar{p} = \bar{B}\sum_k \frac{f_k p_k}{B_k}, \quad \bar{B} = \left(\sum_k \frac{f_k}{B_k}\right)^{-1}, \tag{7}$$

which are derived with the assumption that the changes in material states are small and adiabatic.

Some of the definitions are modified to achieve more robust behavior for real materials encountered in applications (particularly with regard to phase change, fracture and spall). Key among these modifications is the definition of the algorithmic bulk modulus,

$$\tilde{B}_k := \rho_k c_k^2\left(1 + 0.1\frac{\ell}{c_k\Delta t}\right) + 0.1\rho_{0,k}c_{0,k}^2, \tag{8}$$

with $c_k > 0$ the adiabatic material sound speed and $\ell > 0$ a characteristic length scale for the element(cell). This borrows greatly from the definition used by Tipton and has similarity to artificial viscosity. A reference state $\{\rho_{0,k}, c_{0,k}\}$ is used to keep this definition well behaved in material phase space.

ALEGRA supports a material called 'void,' which is the absence of material and has density, energy and pressure of identically zero. Finally, this method is not applied when the material is in tension ($p_k < 0$) or when the element (cell) contains void.

Should these equations be integrated via a forward Euler algorithm with a time step chosen by the stability criteria for the hydrodynamic integrator, the result can be unconditionally unstable. The desire is that the code's time step not be negatively impacted by this algorithm. We have conducted a stability analysis using the classical methods for systems of ODEs. The eigenvalues can become unbounded in two cases, if the volume fraction goes to zero, $f_k \rightarrow 0$ or if the ratio of bulk moduli deviates significantly from one. We have incorporated this analysis in choosing a stable volume fraction or energy update. It is notable that the constant volume fraction method has no extra stability restrictions beyond those imposed by the hydrodynamic integration scheme. While the stability limiter defined below may seem heuristic, its form was chosen to satisfy the detailed stability analysis, which guided our choices.

Limiters for the updates have been used by others [17], but never based on the above stability analysis. This analysis is incorporated into the design of limiters to successfully integrate these equations. In a nutshell, the limiter selects a linear combination of the isentropic update described above with the stable constant volume update resulting in a robust, stable method. Additional adjustments were made to the constants to achieve a better behavior in a large suite of production calculations.

Using the forward Euler time stepping, Equation (4) becomes

$$\Delta f_k = f_k \left(\frac{\bar{B}}{\tilde{B}_k} - 1 \right) (\nabla \cdot \mathbf{v}) \Delta t - f_k \frac{\bar{p} - p_k}{\bar{p}}, \tag{9}$$

and Equation (6) is

$$\Delta e_k = \frac{\bar{p}}{\rho_k f_k} \Delta f_k. \tag{10}$$

The limiter produces a stable volume fraction and energy update

$$\xi := \min_k \left(\frac{f_k}{|\Delta f_k|}, \frac{e_k}{|\Delta e_k|}, \frac{\alpha_k}{|\Delta f_k|} \right), \tag{11}$$

with $\alpha_0 = 0.25$, $\alpha_k = \alpha_0 \min_k \phi(f_k)/\gamma_k^2$, where care is taken if the volume fractions become small, with $\phi(f_k) = \tanh(30 f_k)$, and if the materials are stiff, with $\gamma_k = B_k/p_k$. The update is then $f_k^{n+1} := f_k^n + \xi \Delta f_k$ and $e_k^{n+1} := e_k^n + \xi \Delta e_k$. One last step insures that the volume fractions are physical, $f_k := \max(f_k, 10^{-15})$, and the volume fractions are space filling, $f_k := f_k/\sum f_k$. Figure 1 shows the regions where the stability of the multi-material algorithm is compromised. Where the quantity ξ is non-zero, the full isentropic multi-material update would be unstable without limiters. In these regions, the amount of hybridization with the simpler constant volume fraction algorithm is shown. We might note that the more recent methodology using an (approximate) Riemann solver [15] could provide relief from the necessity of our approach.

The final portion of the algorithm used modifies the temperature of the mixture. We have found that this method tends to cause significant heating of gases when mixed with solids under compression. The temperatures are high enough that one would reasonably expect the materials to be radiatively coupled. To accomplish the physically reasonable behavior of the material, we have implemented a temperature relaxation method based on radiative energy transfer. In order for the method to conserve energy, all materials must relax with identical coupling coefficients.

The temperature relaxation equation used is

$$m_k \frac{\mathrm{d}e_k}{\mathrm{d}t} = \frac{C_{v,k}}{\ell f_k} \sigma(\bar{T}^4 - T_k^4) \approx \frac{4 T_{\max}^3 \sigma C_{v,k}}{\ell f^k} (\bar{T} - T_k), \tag{12}$$

where

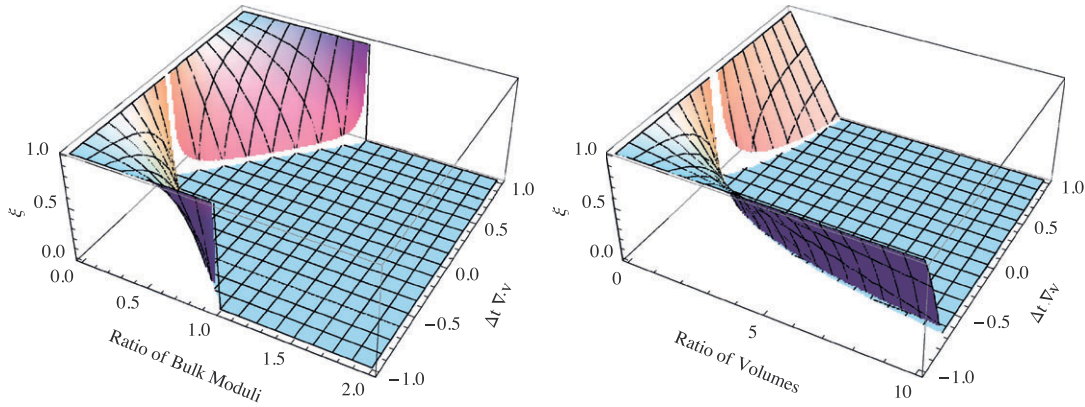$$\bar{T} := \frac{\sum_k m_k C_{v,k} T_k}{\sum_k m_k C_{v,k}}, \tag{13}$$

Figure 1. The surfaces shown provide an analytical expression for the amount of constant volume update needed to provide stability for the algorithm. The left figure shows the linear combination coefficient ξ in terms of the compression of the cell, and the ratio of Bulk moduli and a vanishing ratio of volumes. The right figure shows ξ as a function of the ratio of volumes, and compression, for an effective adiabatic index ($\gamma = -V/p\,\partial p/\partial V_S$) of 4. (a) ξ versus Bulk moduli ratio and compression and (b) ξ versus volume fraction ratio and compression.

thus ensuring that $\sum_k m_k \Delta e_k = 0$. Also, σ is the *Stefan–Boltzmann* constant, ℓ is as before a characteristic length scale for the cell being evolved, $C_v > 0$ is specific heat capacity per unit mass, and $T_{\max} := \max_k(T_k)$ where T_k is the material temperature. As before with the volume fraction and energy updates, a limiter is employed to assure stability; $e_k^{n+1} := e_k^n + \eta \Delta e_k$, where the limiter is $\eta = \min_k (0.25, |e_k/\Delta e_k|)$.

The resulting method has been tested using a broad spectrum of problems ranging from single cell tests for basic functionality to large-scale applications. At the end of this paper, the method is demonstrated on a large-scale application. First, in this section the method will be demonstrated in a fairly standard manner on a modified version of Sod's shock tube [17, 20]. The modified version of the shock tube uses the same density (8 to 1) and pressure (10 to 1) ratios, but uses different ideal gases to the right and left of the interface. On the left side of the interface, $\gamma = 1.8$ and on the right side, $\gamma = 1.2$. In this test, the problem is solved in Lagrangian coordinates and the interface between the two materials is contained in a single cell (element) and evolves in large part through the action of the algorithm described above.

One can test the various aspects of the algorithm including the approach to an equilibrium pressure, or in the case of the thermal algorithm an approach to thermal equilibrium. Initially, we will simply test the basic performance of the method in terms of the pressure, temperature, volume fraction, and overall solution. Figure 2 shows how the method performs for these variables.

3. MULTI-MATERIAL REMAP STEP

The remap is an essential portion of an ALE algorithm. The remap moves data from the grid deformed by the Lagrangian step to another mesh presumably chosen to have a better character than the Lagrangian grid. Often the remap will move data back to the grid at the start of the step. If this is done for each step of the integration, the calculation is effectively Eulerian in character.

Two principal types of error are expected during the remap step, oscillations in quantities and numerical diffusion. These types of errors are played against each other, with numerical diffusion controlling potential oscillations through the action of limiters usually enforcing a monotonicity principle. We will engage in this trade space as well, but include the presence of a material interface in the balance between diffusion and oscillations.

Without a doubt, the presence of a material interface can produce inaccuracies in the production of high-order approximations in the remap. Typically, the remap works by computing overall
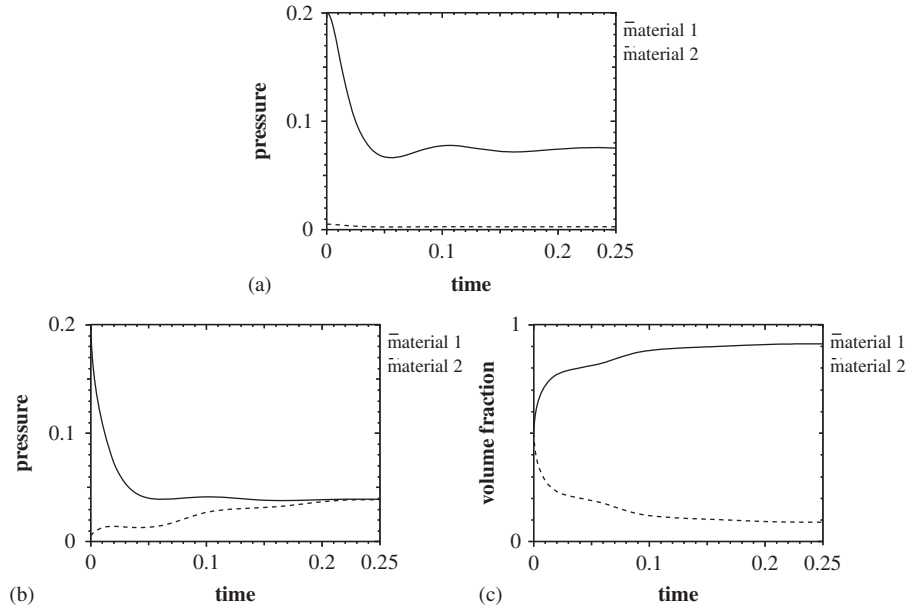
Figure 2. Performance of the multi-material Lagrangian closure on the modified Sod shock tube problem. The constant volume pressure evolution is shown in (a); the volume fractions remain fixed at 50% for all time. (b) Shows the equilibration of the pressure (over 118 steps), and (c) displays the large change in the volume fraction with the isentropic multi-material closure. The overall model uses 40 mesh cells and runs to a non-dimensionless time of 0.25. The quantities plotted are the values in the cell that contains the discontinuous interface between the materials. (a) Constant volume fraction pressure versus time; (b) isentropic pressure versus time; and (c) isentropic volume fraction versus time.

volumetric fluxes of material which are partitioned by the relative volume of each constituent material. Quantities such as density, energy or other material intensive variables are scaled by the volume flux. These quantities are reconstructed or differenced independently from the material interface and care must be taken to produce an accurate and high-fidelity representation of quantities in the vicinity of the interface. Other variables such as velocity are not material dependent, but rather continuous. We have found that these values also need to be aware of the material interface.

In ALEGRA, the remap is accomplished on a locally logically rectangular grid in a local coordinate system either in terms of volume or mass coordinates. The algorithm is dimensionally split in nature using Strang splitting [21] to achieve second-order accuracy in a linear sense. In the past, ALEGRA has used a Van Leer type method [3] (a monotonicity limited Fromm scheme) for remap. The variables are all updated in a conservation form

$$V_j w_j^{n+1} = V_j w_j^n - \Delta t (A_{j+1/2} w_{j+1/2}^{n+1/2} - A_{j-1/2} w_{j-1/2}^{n+1/2}), \tag{14}$$

where w_j is the variable being remapped, V is the element volume and A is the area through which the remap is occurring. This algorithm is described in brief here starting with the linear reconstruction

$$w(\theta) = p_0 + p_1 \theta, \quad p_0 = \bar{w}_j, \quad p_1 = s_j, \quad \theta \in [-1/2, 1/2], \tag{15}$$

which evaluates to the following accurate value:

$$w\left(\pm\frac{1}{2} - \frac{v}{2}\right) = w_{j\pm1/2}^{n+1/2} = p_0 + \frac{1}{2} p_1 (\pm 1 - v); \quad v = \frac{\Delta t}{V_j} (\mathbf{v} \cdot \mathbf{n})_{j\pm1/2} A_{j\pm1/2}. \tag{16}$$

The accurate slope is

$$s_j = \frac{(\Delta\xi_{j-1})^2 w_{j+1} + [(\Delta\xi_{j-1})^2 - (\Delta\xi_{j+1})^2] w_j - (\Delta\xi_{j+1})^2 w_{j-1}}{\Delta\xi_{j-1}\Delta\xi_{j+1}(\Delta\xi_{j-1} - \Delta\xi_{j+1})},$$

where $\Delta\xi_j$ the change in volume or mass coordinate. The slope, s_j, is limited as follows: $s_- = w_j - w_{j-1}$, $s_+ = w_{j+1} - w_j$, $Q^* = 2 \cdot \text{median}(0, s_-, s_+)$, and finally $s_j := \text{median}(0, s_j\Delta\xi_j, Q^*)/\Delta\xi_j$.[§] This method is formally second-order accurate for smooth data.

This method has been replaced by a formally third-order accurate method (for smooth data) based loosely on the PPM algorithm [22], but on a more compact stencil. The method can be described similarly to the legacy method beginning with the quadratic reconstruction

$$w(\theta) = p_0 + p_1\theta + p_2\theta^2, \tag{17}$$

with

$$\left.\begin{aligned} p_0 &= \tfrac{3}{2}w_j - \tfrac{1}{4}\left(w_{j-1/2} + w_{j+1/2}\right) \\ p_1 &= w_{j+1/2} - w_{j-1/2} \\ p_2 &= 3\left(w_{j-1/2} + w_{j+1/2}\right) - 6w_j \end{aligned}\right\}. \tag{18}$$

The monotonicity is computed in two steps using $w_{j\pm1/2} := \text{median}\left(w_j, w_{j\pm1/2}, w_{j\pm1}\right)$ in the first step, and $w_{j\pm1/2} := \text{median}\left(w_j, w_{j\pm1/2}, 3w_j - 2w_{j\mp1/2}\right)$ in the second. The value fluxed is computed using

$$w_{j\pm1/2}^{n+1/2} = p_0 + p_1\left(\pm\frac{1}{2} - \frac{1}{2}v\right) + p_2\left(\frac{1}{4} \mp \frac{1}{2}v + \frac{1}{3}v^2\right). \tag{19}$$

The key is to compute the edge values, $w_{j\pm1/2}$, so that they are third-order accurate and appropriately bounded. This step also obviates the need for the first bounding step (the first median function) in the PPM monotonicity algorithm above. We introduce a form for the third-order method which insures that the value lies between the adjacent values of w. The algorithm has two steps, first computing a second-order value, and then correcting that value to third order. Thus,

$$w_{j\pm1/2} = \underbrace{\left(\frac{\Delta\xi_j w_{j\pm1} + \Delta\xi_{j\pm1} w_j}{\Delta\xi_j + \Delta\xi_{j\pm1}}\right)}_{\text{second order}} - \underbrace{(\delta_{j\pm1/2}w - \delta_{j\mp1/2}w)}_{\text{third order correction}}, \tag{20}$$

where

$$\delta_{j+1/2}w = \text{minmod}\left(\frac{w_{j+1} - w_j}{\Delta\xi_j + \Delta\xi_{j+1}}, 4\frac{w_j - w_{j-1}}{\Delta\xi_j + \Delta\xi_{j-1}}\right),$$

and

$$\delta_{j-1/2}w = \text{minmod}\left(\frac{w_j - w_{j-1}}{\Delta\xi_j + \Delta\xi_{j-1}}, 4\frac{w_{j+1} - w_j}{\Delta\xi_j + \Delta\xi_{j+1}}\right).$$

The limiter on the edge differences can be changed to affect a more dissipative method by changing the coefficient four in the minmod limiter for computing $\delta_{j\pm1/2}$ to a lower value. We choose a coefficient of two and denote this method as the safe PPM method. If the minmod limiter does not modify the edge differences, the approximation produces the classical third-order approximation for the edge values on this three point stencil.

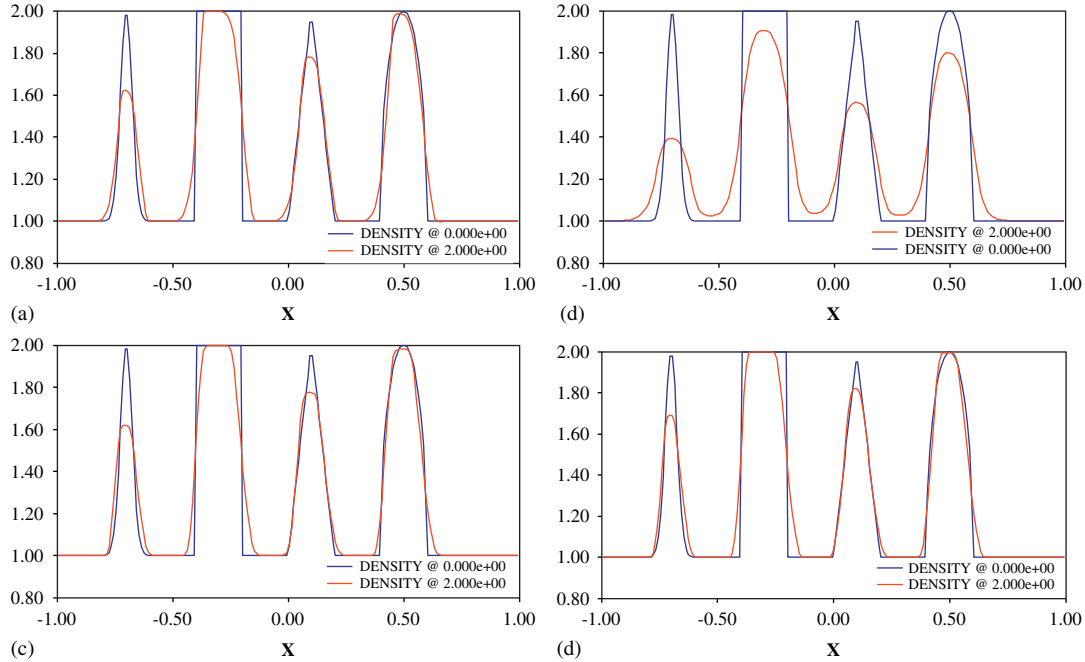[§]The median function takes three arguments and returns the argument bounded by the other two.

Figure 3. These figures demonstrate the relative performance of the remapping methods discussed here. The methods are tested on the four wave profile consisting of a Gaussian, square wave, triangular wave, and ellipse. The methods from left to right, top to bottom are the Van Leer, minmod, safer third-order and regular third-order method. The solution is shown after a single period using 200 cells at a CFL number of 0.10. For this problem, the relative error magnitude in the L_1 norm with the third order being 1.0 are safer (1.10), Van Leer (1.23), and minmod (2.74): (a)Van Leer; (b) MinMod; (c) safer third order; and (d) regular third order.

The adaptive differencing chosen makes use of these approximations along with donor cell or first-order upwind and one additional method described next. This is the classical second-order minmod method [23, 24], which is the most dissipative second-order monotone (TVD) method. The method is quite simple,

$$w(\theta) = w_j + s_j; \quad s_j = \Delta\xi_j \text{minmod}\left(2\frac{w_j - w_{j-1}}{\Delta\xi_j + \Delta\xi_{j-1}}, 2\frac{w_{j+1} - w_j}{\Delta\xi_j + \Delta\xi_{j+1}}\right). \quad (21)$$

The time-averaged value used for the fluxes is identical to the legacy (van Leer) value with this modified slope.

The candidate methods are tested on a standard waveform test [25, 26]. The waveform contains a Gaussian, square wave, triangular wave, and ellipse. Figure 3 shows the results for each method. The third-order method is easily the best, and the safer third-order method and Van Leer are similar although the third-order method is marginally better. The minmod method is much more diffusive, but less dissipative than donor cell. Separate testing with a C_∞ function (the Gaussian) demonstrates that the formal order of the methods in L_1 are as follows: minmod—second order, Van Leer—2nd order, safer third-order—3rd order, and regular third-order—3rd order.

The key to the method is to modify the choice of differencing in the vicinity of a material interface. Early results from applications indicated that the choice of using donor cell in the mixed cells [18] while producing more physical states was too dissipative and negatively impacted simulation results. Obviously in seeking a better approach the results must be physically admissible, while not inducing too much dissipation. In the course of this study, we chose the most dissipative second-order method instead of donor cell, the minmod method for the differencing in mixed cells. The minmod method chooses the shallowest slope and will naturally choose to difference within
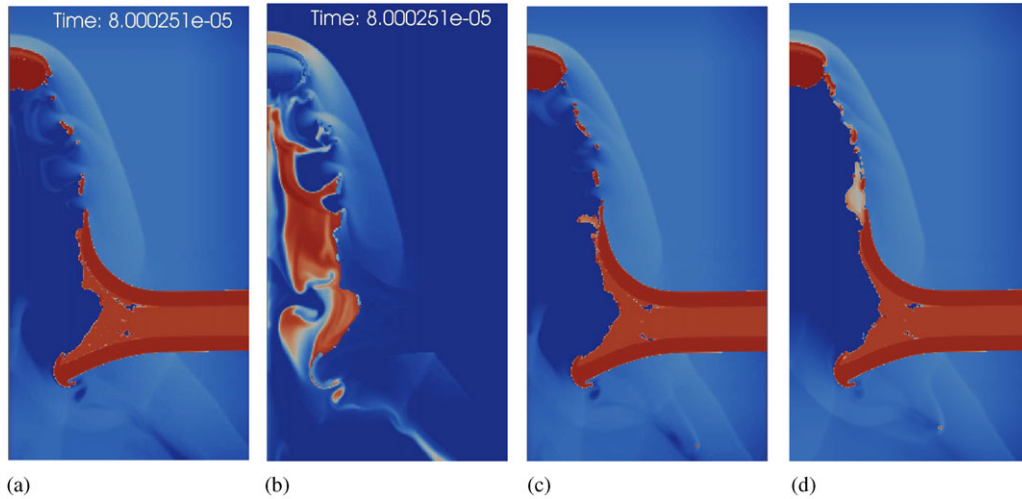
Figure 4. This shows the relative differences in a calculation of a complex satellite shield as a function of the multi-material closure of the Lagrangian step. The methods are shown from left to right as the constant volume, basic isentropic, isentropic with thermal radiation and isentropic with thermal equilibration. The largest differences are the phase of the solid materials which are largely a function of the material temperature. The model uses a mesh of 203 by 402 cells in the radial and axial directions. The colors are scaled by the logarithm of the density. (a) Constant volume fraction; (b) basic isentropic; (c) isentropic with radiative coupling; and (d) isentropic with thermal equilibration.
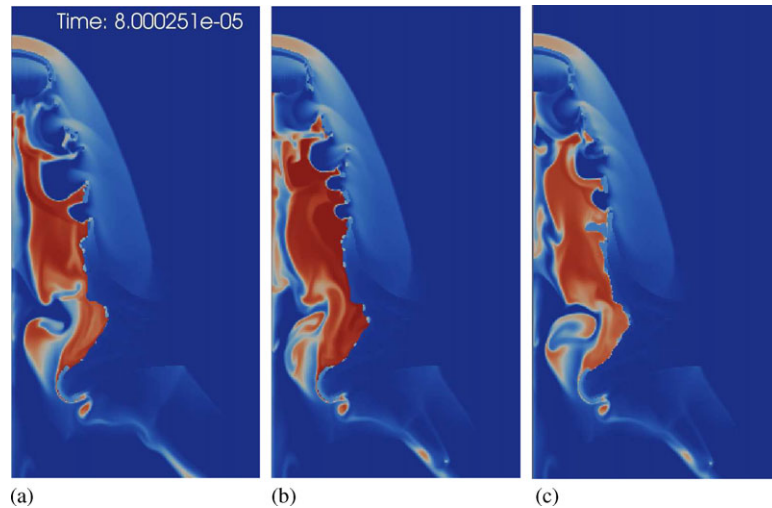

Figure 5. This figure shows the differences in the air temperature as a function of the multi-material closure of the Lagrangian step. The methods shown from left to right are the constant volume, basic isentropic and isentropic with thermal radiation. The model uses a mesh of 203 by 402 cells in the radial and axial directions. The colors are scaled by the logarithm of the temperature. (a) Constant volume fraction; (b) basic isentropic; and (c) isentropic with radiative coupling.

the same material as the mixed cell rather than into the empty cell, but do so safely due to the limiter employed.

In studying methods near interfaces, we also found that it was advantageous to modify the high-order method adjacent to interfaces making the change to minmod less abrupt. The PPM-safe method was chosen as a slightly more dissipative version. Finally, if a material interface is isolated from other cells with that material it is treated with donor cell.
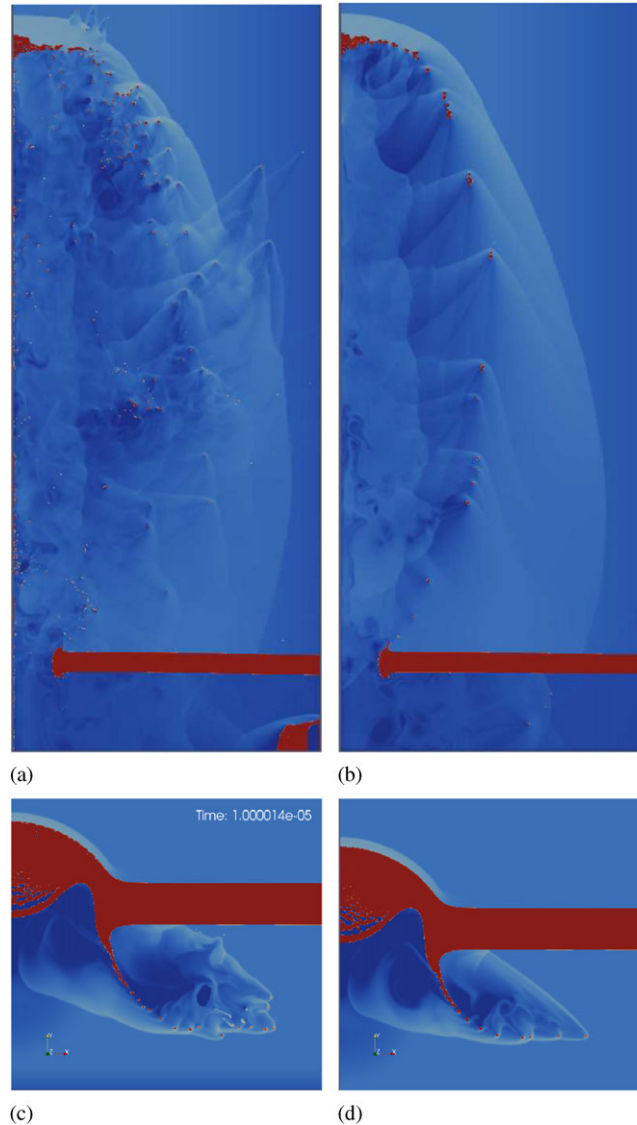
Figure 6. This figure shows the fundamental differences between the legacy remap and the new adaptive remap for a simple 'Whipple' satellite shield. The upper two figures show the late time behavior of the calculations at the medium grid resolution with legacy remap on the left and the new remap on the right. The bottom figures show the fine grid resolution calculations at the time when the Legacy method fails due to a too small time step size. The medium mesh uses 203 cells in the radial and 675 cells in the axial direction. The fine mesh uses 406 cells in the radial and 1347 cells in the axial direction. The colors are scaled by the logarithm of the density. (a) Medium grid with legacy remap; (b) medium grid with new remap; (c) fine grid with legacy remap; and (d) fine grid with new remap.

4. RESULTS

Two problems are used to demonstrate the results associated with protecting satellites from damage from micrometeorites through shielding. The first problem has layers of light materials impacted by an iron sphere traveling at 2 km per second. The layered shield has lexan sandwiched between aluminum plates. The second problem is simpler with a plate of iron impacted by a faster iron sphere traveling at 4 km per second.

The first example shows the differences between the multi-material Lagrangian closure algorithms. All the calculations utilize the adaptive remap rather than the older standard method
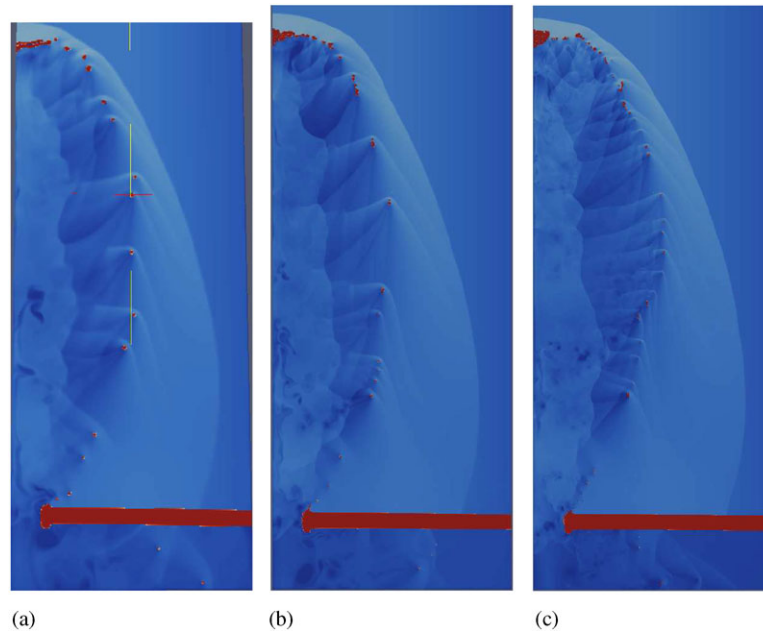
Figure 7. This figure shows a mesh refinement sequence for the Whipple shield from coarse resolution on the left to the fine resolution on the right. For this problem, the adaptive method is convergent under mesh refinement; the legacy remap method appears to be divergent. The coarse mesh uses 101 cells in the radial and 339 cells in the axial direction. The medium mesh uses 203 cells in the radial and 675 cells in the axial direction. The fine mesh uses 406 cells in the radial and 1347 cells in the axial direction. The colors are scaled by the logarithm of the density. (a) Coarse grid; (b) medium grid; and (c) fine grid.

using Van Leer's method for all cells. While the large-scale evolution of the damage to the shield is nearly identical, the details of the damage to the shield vary substantially. Figure 4 shows a plot of the logarithm of density; the major differences are associated with the lexan portion of the shield. The lexan is impacted substantially by the temperature of the material while it evolves. The temperature of the air is shown in Figure 5. The isentropic algorithm results in much greater compression of the air mixed with solids hence greater pdV heating and temperature. The radiative coupling reduces the temperatures achieved in the air to roughly those computed with the constant volume method. If the thermal relaxation is modified to bring the temperatures into equilibrium, the temperatures in the air are dramatically reduced. The consequence of the energy transfer in these algorithms is the melting of materials such as lexan in the problem and a substantial influence on the state of the debris computed. We note that none of these results is distinctly identified as being 'better.' The determination of the best result would depend upon experimental data to validate the effective modeling associated with our closure methodology.

The second example involves the Whipple shield in order to demonstrate the differences in the advective remap differencing. For all the calculations shown here the isentropic multi-material method is used in the Lagrangian step. The legacy remap using Van Leer produces decidedly unphysical results as the plume of debris produced by the calculations is not consistent with experimental observations. This behavior is precisely the issue that dictated our adaptive approach to remap. The breakup of the spall bubble is not associated with the Van Leer method, but the use of the higher-order remap across the material interface. This is shown in Figure 6 where the legacy method is compared with our new adaptive remap. The new adaptive remap produces a bow shock and debris pattern consistent with the observed behavior. Furthermore, when the grid is refined the legacy method does not produce a viable calculation because the time step size drops to the point where the calculation no longer progresses. In the lower right-hand side of the image showing the medium resolution legacy remap results, some debris has expanded to a

grossly unphysical state—the newer remap does not produce this pathological behavior. The newer method produces a full calculation which can be demonstrated to be convergent for a number of quantities of interest, such as the speed and shape of the debris envelope. This character is shown in Figure 7.

5. CLOSURE

This paper has described the development, implementation and analysis of two new methods deployed in the ALE code ALEGRA. The first method provides a subcell evolution of materials in the Lagrangian step of the ALE algorithm. A stability analysis has been utilized to insure that the method is stable and robust. The second algorithm is an adaptive remap method where the non-linear differencing is chosen based on the proximity of the cell to a material interface. This method has been demonstrated to have a large impact on the results and robustness of calculations using ALEGRA.

REFERENCES

1. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**(3):227–253.
2. Boris JP, Book DL. Flux-corrected transport I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics* 1973; **11**:38–69.
3. van Leer B. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics* 1977; **23**:276–299.
4. Taylor LM, Flanagan DP. PRONTO-2D: a two-dimensional transient solid dynamics program. *Technical Report SAND86-0594*, Sandia National Laboratories, Albuquerque, NM, March 1987.
5. Taylor LM, Flanagan DP. PRONTO3D: a three-dimensional transient solid dynamics program. *Technical Report SAND87-1912*, Sandia National Laboratories, Albuquerque, NM, March 1989.
6. Bell RL *et al.* CTH user's manual and input instructions, version 4.00. *Technical Report*, Sandia National Laboratories, Albuquerque, NM, April 1999.
7. Perry JS, Budge KG, Wong MKW, Trucano TG. RHALE: a 3-D MMALE code for unstructured grids. In *Advanced Computational Methods for Material Modeling*, AMD-Vol. 180/PVP-Vol. 268. ASME, 1993; 159–174.
8. Budge KG, Peery JS. RHALE: a MMALE shock physics code written in C++. *International Journal of Impact Engineering* 1993; **14**(1–4):107–120.
9. Loubere R, Maire P-H, Shashkov M, Breil J, Galera S. Reale: a reconnection-based arbitrary-Lagrangian–Eulerian method. *Journal of Compuational Physics* 2010; **229**(12):4724–4761.
10. Knupp P, Margolin L, Shashkov M. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Compuational Physics* 2002; **176**(1):93–128.
11. Kucharik M, Shashkov M, Wendroff B. An efficient linearity-and-bound-preserving remapping methods. *Journal of Compuational Physics* 2003; **188**(2):462–471.
12. Shashkov M, Wendroff B. The repair paradigm and application to conservation laws. *Journal of Compuational Physics* 2004; **198**(1):265–277.
13. Tipton RE. *The CALE User's Manual*. Version 901101. Lawrence Livermore National Laboratory, Livermore, CA 1990.
14. Tipton RE. CALE mixed zone pressure relaxation model. Private communication, 1989.
15. Barlow A. A new Lagrangian scheme for multimaterial cells. *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering. ECCOMAS Computational Fluid Dynamics Conference*, AIAA Paper 93-3345, Swansea, U.K., 2001.
16. Wilkins ML. Use of artificial viscosity in multidimensional fluid dynamic calculations. *Journal of Computational Physics* 1980; **36**:281–303.

17. Shashkov M. Closure models for multimaterial cells in arbitrary Lagrangian–Eulerian hydrocodes. *International Journal for Numerical Methods in Fluids* 2008; **56**(8):1497–1504. DOI: 10.1002/fld.1574.
18. Miller GH, Puckett EG. A high-order Godunov method for multiple condensed phases. *Journal of Computational Physics* 1996; **128**:134–164.
19. Colella P, Glaz HM, Ferguson R.E. Multifluid Algorithms for Eulerian Finite Difference Methods, In preparation.
20. Sod G. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics* 1978; **27**:1–31.
21. Strang G. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis* 1968; **5**:506–517.
22. Colella P, Woodward P. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics* 1984; **54**:174–201.
23. Harten A. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* 1983; **49**:357–393. Reprinted in Volume 135 Number 2, pp. 260–278, August 1997.
24. Sweby PK. High-resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal of Numerical Analysis* 1984; **21**:995–1011.
25. Suresh A, Huynh HT. Accurate monotonicity-preserving schemes with Runge-Kutta time stepping. *Journal of Computational Physics* 1997; **136**:83–99.
26. Rider WJ, Greenough JA, Kamm JR. Accurate monotonicity- and extrema-preserving methods through adaptive nonlinear hybridizations. *Journal of Compuational Physics* 2007; **225**(2):1827–1848.